In [1]:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

```

In [2]:

```
url = 'https://www.themoviedb.org/movie?page='
page_url = 'https://www.themoviedb.org'
```

In [3]:

```
content=requests.get(url,headers={'User-Agent':'Mozilla/5.0'}).text
```

In [ ]:

```

```

In [4]:

```
## movie_url of 1st page
soup = BeautifulSoup(content,'lxml')
movie_url_1st_page = []
movie_name_lst = []
data = soup.find_all('div',class_ = 'card style_1')
for i in data:
    movie_code = i.a['href']
    movie_url_1st_page.append(page_url+movie_code)
    names = i.a['title']
    movie_name_lst.append(names)

```

In [5]:

```
#movie_url_1st_page

```

In [6]:

```
#movie_name_lst
```

```
movie_name_lst =[]
director_lst = []
gen_lst = []
run_time = []
relese_lst = []
raiting_lst = []

for link in movie_url_1st_page:
    content=requests.get(link,headers={'User-Agent':'Mozilla/5.0'}).text

    soup = BeautifulSoup(content,'lxml')
```

```
    data = soup.find_all('div',class_ = 'header_poster_wrapper true')

    raiting = soup.find('div',class_ = 'user_score_chart')['data-percent']
    raiting_lst.append(raiting)


    relese_date = soup.find('span',class_ = 'release')

    director = soup.find('li',class_ = 'profile').a.text
    director_lst.append(director)

    val = soup.find('span',class_ = 'genres').text.split()

    gen_lst.append(val)


    runtime = soup.find('span',class_ = 'runtime').text.split()
    run_time.append(runtime)

    movie_data_dic = {
        'Movie Name': movie_name_lst,
        'Director': director_lst,
        'Release Date' : relese_lst,
        'Run Time': run_time,
        'Genres': gen_lst,
        'Raiting' : raiting_lst,
        'Movie_link': movie_url_1st_page

    }
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```

```

In [ ]:

```

```

```

```

In [7]:

```
url_lst = []
for u in range(0,501):
    url_lst.append(url+str(u))
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [8]:

```
movie_url_all_pages = []
movie_name_lst = []

for link in url_lst:
    content=requests.get(link,headers={'User-Agent':'Mozilla/5.0'}).text
    soup = BeautifulSoup(content,'lxml')
    data = soup.find_all('div',class_ = 'card style_1')

    for i in data:
        movie_code = i.a['href']
        movie_url_all_pages.append(page_url+movie_code)
        names = i.a['title']
        movie_name_lst.append(names)

```

In [ ]:

```
1
```

In [9]:

```
1 len(movie_name_lst)
```

Out[9]:

```
10000
```

In [10]:

```
director_lst = []
gen_lst = []
run_time = []
release_lst = []
raiting_lst = []

for link in movie_url_all_pages:
    content=requests.get(link,headers={'User-Agent':'Mozilla/5.0'}).text


    soup = BeautifulSoup(content,'lxml')
    data = soup.find_all('div',class_ = 'header_poster_wrapper true')

    raiting = soup.find('div',class_ = 'user_score_chart')['data-percent']
    raiting_lst.append(raiting)


    release_date = soup.find('span',class_ = 'release').text.split()[0]
    release_lst.append(release_date)

    director = soup.find('li',class_ = 'profile')
    if director is not None:
            director=(director.p.text)
    director_lst.append(director)

    val = str(soup.find('span',class_ = 'genres').text)
    genres = val.replace('\n','')
    gen_lst.append(genres)

    runtime = soup.find('span',class_='runtime')
    #if runtime is not None:
        #runtime=(runtime.text.strip())
        #run_time.append(runtime)
    #runtime = (soup.find('span',class_ = 'runtime'))
    time = runtime.replace('\n','')
    if runtime is not None:
         runtime=runtime.text


    run_time.append(runtime)

    movie_data_dic = {
        'Movie Name': movie_name_lst,
        'Raiting' : raiting_lst,
        'Release Date' : release_lst,
        'Run Time': run_time,
        'Genres': gen_lst,
        'Director': director_lst,
        'Movie_link': movie_url_all_pages

     }

```

```
---------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
~\AppData\Local\Temp/ipykernel_11220/193167220.py in <module>
```

```
     33         #run_time.append(runtime)
     34     #runtime = (soup.find('span',class_ = 'runtime'))
---> 35     time = runtime.replace('\n','')
     36     if runtime is not None:
     37         runtime=runtime.text

TypeError: 'NoneType' object is not callable
```

In [12]:

```
1  len(gen_lst)
```

Out[12]:

```
2536
```

In [13]:

```
1  df = pd.DataFrame(movie_data_dic)
2  df
```

```
---------------------------------------------------------------------------
--
ValueError                                Traceback (most recent call las
t)
~\AppData\Local\Temp/ipykernel_5696/1009747518.py in <module>
----> 1 df = pd.DataFrame(movie_data_dic)
      2 df

~\anaconda3\lib\site-packages\pandas\core\frame.py in __init__(self, dat
a, index, columns, dtype, copy)
    612         elif isinstance(data, dict):
    613             # GH#38939 de facto copy defaults to False only in no
n-dict cases
--> 614             mgr = dict_to_mgr(data, index, columns, dtype=dtype,
 copy=copy, typ=manager)
    615         elif isinstance(data, ma.MaskedArray):
    616             import numpy.ma.mrecords as mrecords

~\anaconda3\lib\site-packages\pandas\core\internals\construction.py in di
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```